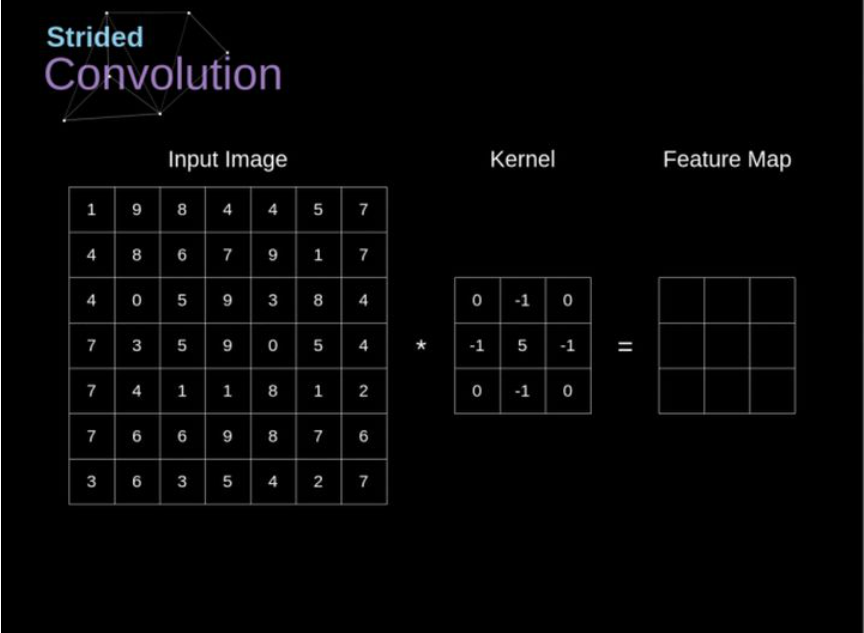
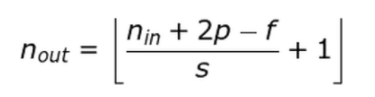
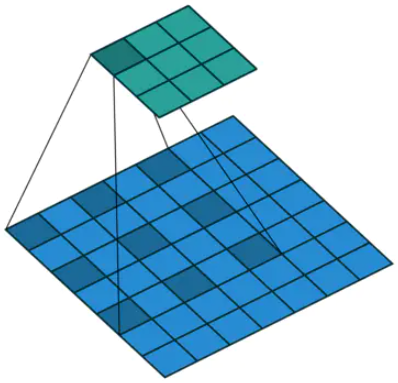
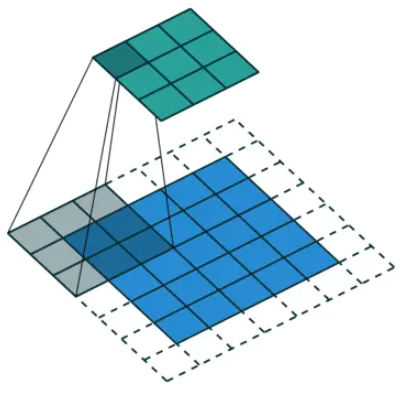
*跨步卷积：*



在之前的例子中，我们总是将卷积核移动一个像素。但是，步长也可以看做是卷积层的一个参数。在图 6 中，我们可以看到，如果我们使用更大的步长，卷积会成为什么样子。在设计 CNN 结构时，如果我们想让接受域有更少的重叠或者想让特征图有更小的空间维度，那么我们可以决定增大步长。考虑到扩充和跨步，输出矩阵的维度可以使用下面的公式计算：



*空洞卷积：*



左图为普通卷积，右图为空洞卷积，并且空洞为2

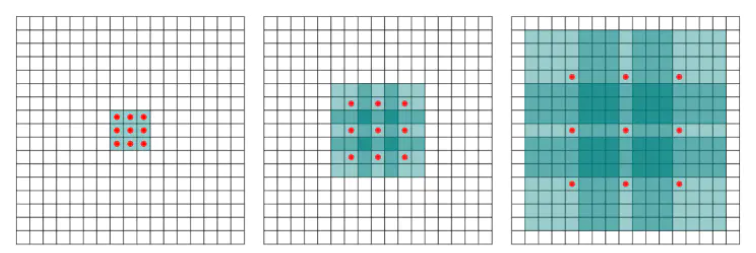
起源：

CNN网络对于其他任务有一些致命的缺陷，较为著名的是 up-sampling和pooling layer

主要问题有：

1. Up-sampling/pooling layer的参数不可学习
2. 内部数据结构丢失，空间层级化信息丢失
3. 小物体信息无法重建(假设有4个pooling layer 则 任何小于 2^4 = 16 pixel的物体信息将理论上无法重建。)

在图像分割领域，图像输入到CNN (典型的网络如FCN)中，FCN先像传统的CNN那样对图像做卷积再pooling，降低图像尺寸的同时增大感受野，但是由于图像分割预测是pixel-wise的输出，所以要将pooling后较小的图像尺寸upsampling到原始的图像尺寸进行预测，之前的pooling操作使得每个pixel预测都能看到较大感受野信息。因此图像分割FCN中有两个关键，一个是pooling减小图像尺寸增大感受野，另一个是upsampling扩大图像尺寸。在先减少再增大尺寸的过程中，肯定有一些信息损失掉了，因此使用dilated conv来减少在尺寸变换中的信息损失。



dilated conv

左图对应3×3的1-diated conv，和普通的卷积操作一样，中图对应3×3的2-dilated conv，实际的卷积kernel size还是3×3，但空洞为1，也就是对于一个7×7的图像patch，只有9个红色的点和3×3的kernel发生卷积操作，其余点略过。也可以理解为kerenl的size为7×7（如果考虑到这个2-dilated conv的前一层是一个1-dilated conv的话，那么每个红点就是1-dilated的卷积输出，所以感受野为3×3，所以1-dilated和2-dilated合起来就能达到7×7的conv），右图是4-dilated conv操作，同理跟在两个1-dilated conv 和2-dilated conv的后面，就能达到15×15的感受野。对比传统的conv操作，3层3×3卷积加起来，stride为1的话，只能达到(kernel - 1) \* layer + 1 = 7的感受野，也就是和层数layer成线性关系，而dilated conv的感受野是指数级的增长。

潜在问题：

空洞卷积的kernel并不连续，也就是说并不是所有的pixel都用来计算了，因此这里将信息看作checker-board的方式会损失信息的连续性，这对pixel-level dense prediction的任务来说是致命的。

我们从 dilated convolution 的设计背景来看就能推测出这样的设计是用来获取 long-ranged information。然而光采用大 dilation rate 的信息或许只对一些大物体分割有效果，而对小物体来说可能则有弊无利了。如何同时处理不同大小的物体的关系，则是设计好 dilated convolution 网络的关键。